LA-UR-19-32693
Approved for public release; distribution is unlimited.

Title: DRACO: An Overview

Author(s): Barber, John L.

Intended for: Document general presentation
Report

Issued: 2019-12-20

(U) DRACO: An Overview

John L. Barber (T-1)

Introduction

The set of partial differential equations solved by DRACO are of the typical reaction-diffusion form:

Time derivatives of temperature, concentration, pressure, etc.

$$\left\{ \begin{aligned} \frac{\partial u}{\partial t} &= \overbrace{\vec{\nabla} \cdot \left(D_u \vec{\nabla} u \right)}^{\text{Diffusion}} + \overbrace{\left(\frac{du}{dt} \right)_{\mathrm{C}}}^{\text{Chemistry}} + \overbrace{\left(\frac{du}{dt} \right)_{\mathrm{S}}}^{\text{Sources/Sinks}} \\ \frac{\partial v}{\partial t} &= \vec{\nabla} \cdot \left(D_v \vec{\nabla} v \right) + \left(\frac{dv}{dt} \right)_{\mathrm{C}} + \left(\frac{dv}{dt} \right)_{\mathrm{S}} \end{aligned} \right.$$

$$\vdots$$

Diffusion coefficients
(May depend on position, time, concentration, etc.)

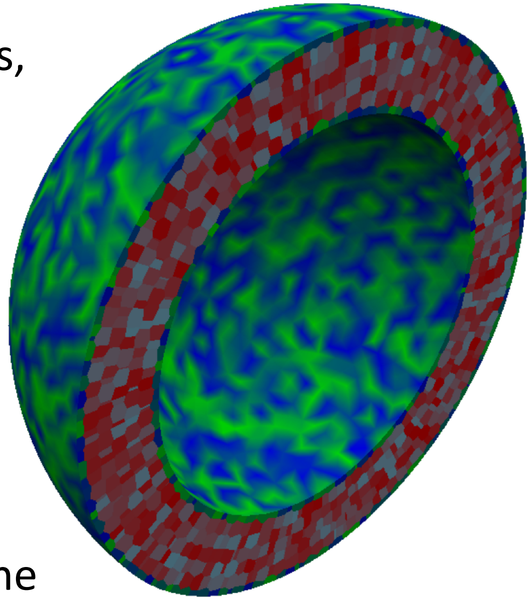
General, nonlinear chemistry

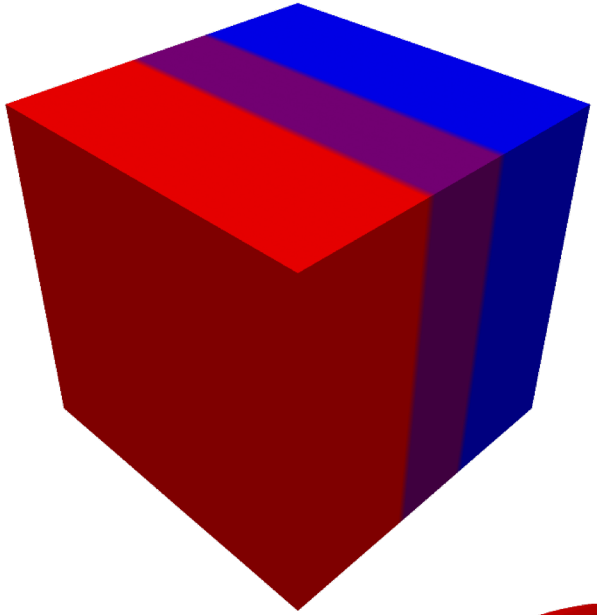
Arbitrary sources/sinks

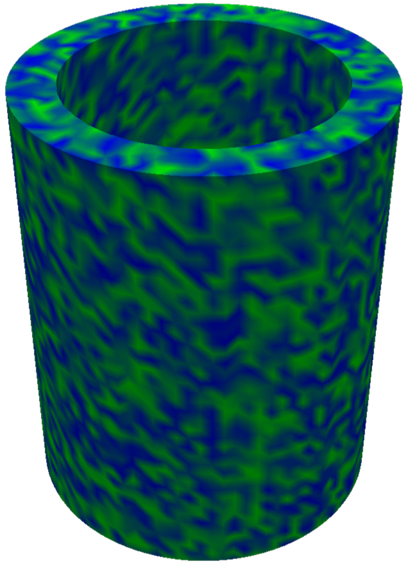
DRACO allows an arbitrary number of **diffusers** *(u, v, …)*, as well as an arbitrary number of **parts** which may communicate (i.e. exchange material) in an any arbitrary way.

Meshing: DRACO "Native" Shapes

DRACO can generate its own meshes on a set of simple shapes such as spheres, cones, boxes, tori, etc.

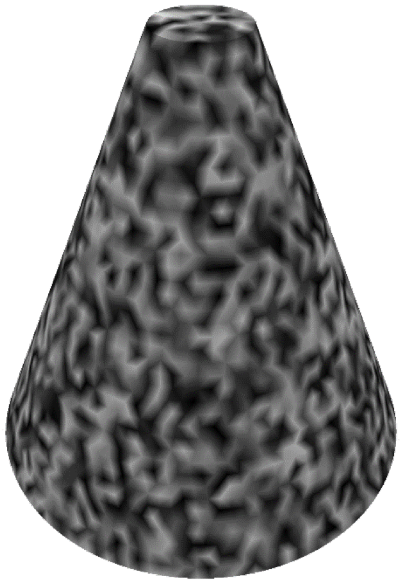


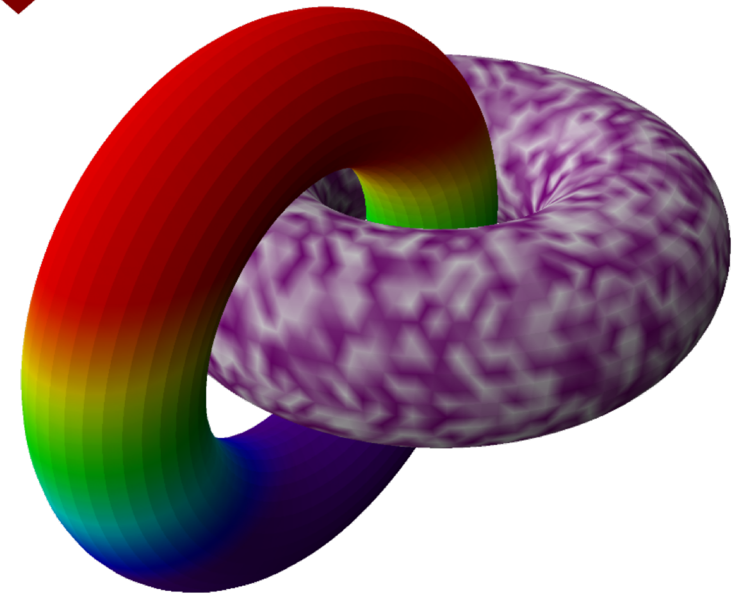




DRACO also performs all of its own ray tracing and image rendering, without relying on any external library or software.

In general, DRACO has been written with the goal of avoiding external libraries or software whenever possible. This was to ensure that it works "out of the box."



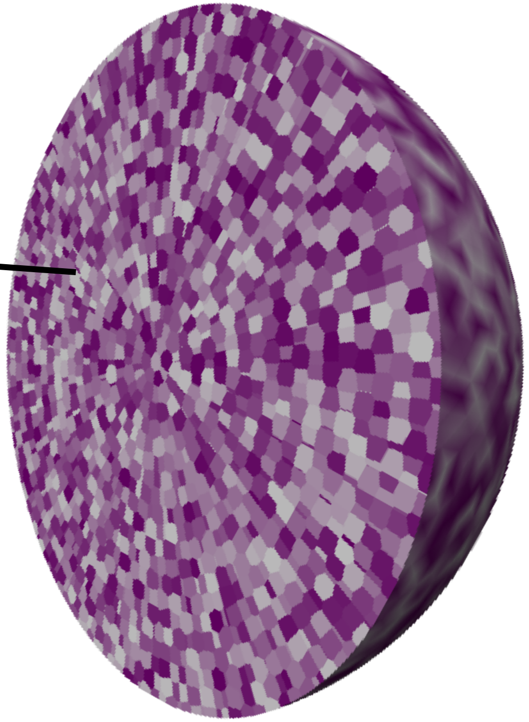


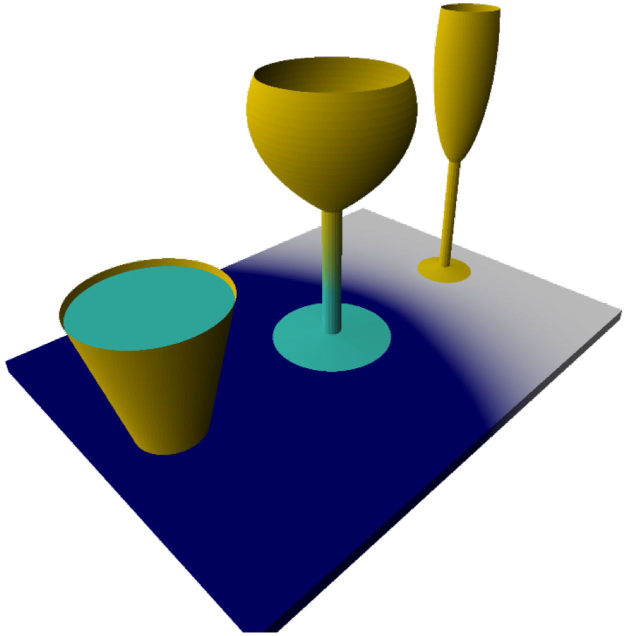
The major exception to this is the use of the Voro++ Voronoi tessellation library, which is available for free from the LBNL website.

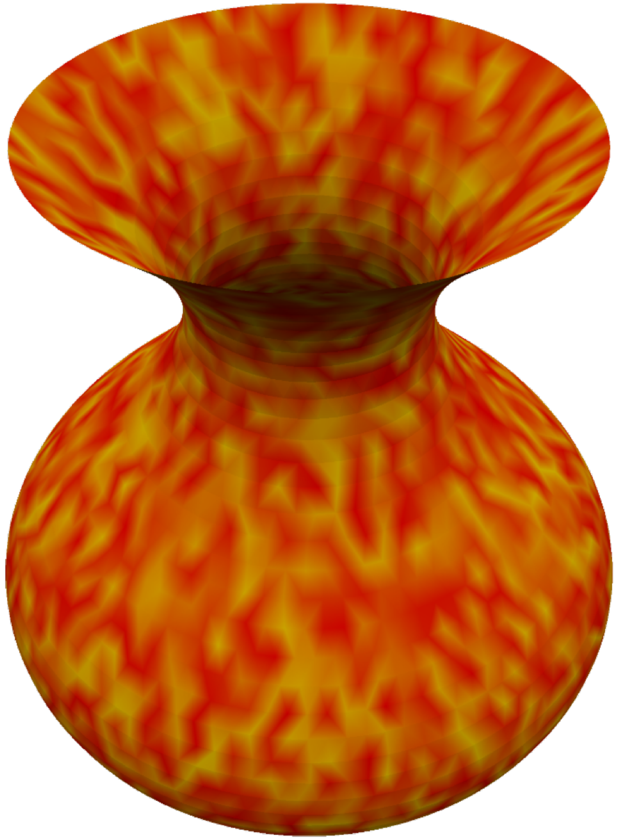
Meshing: Radially-Symmetric Shapes

DRACO also has the capability to mesh arbitrary radially-symmetric shapes using a meshing scheme known as **Fibonacci meshing**.

Fibonacci meshing yields extraordinarily-smooth sampling of “round things,” with even resolution and without “cusp points” or “special meridians.”





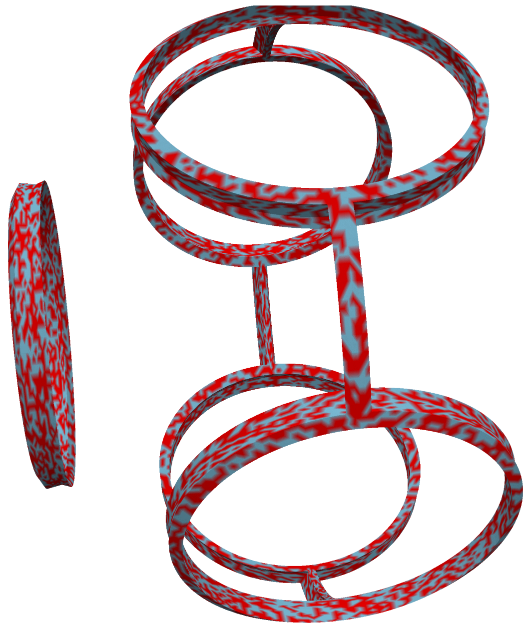


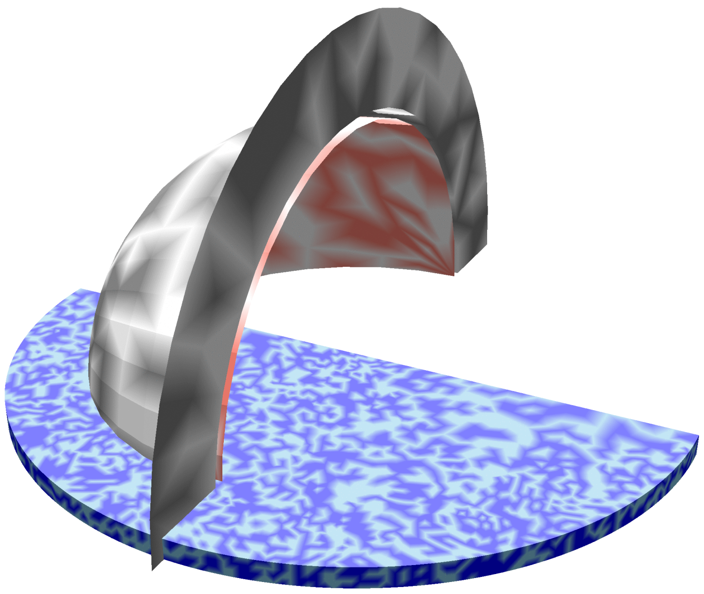
Radially-Symmetric Part Examples

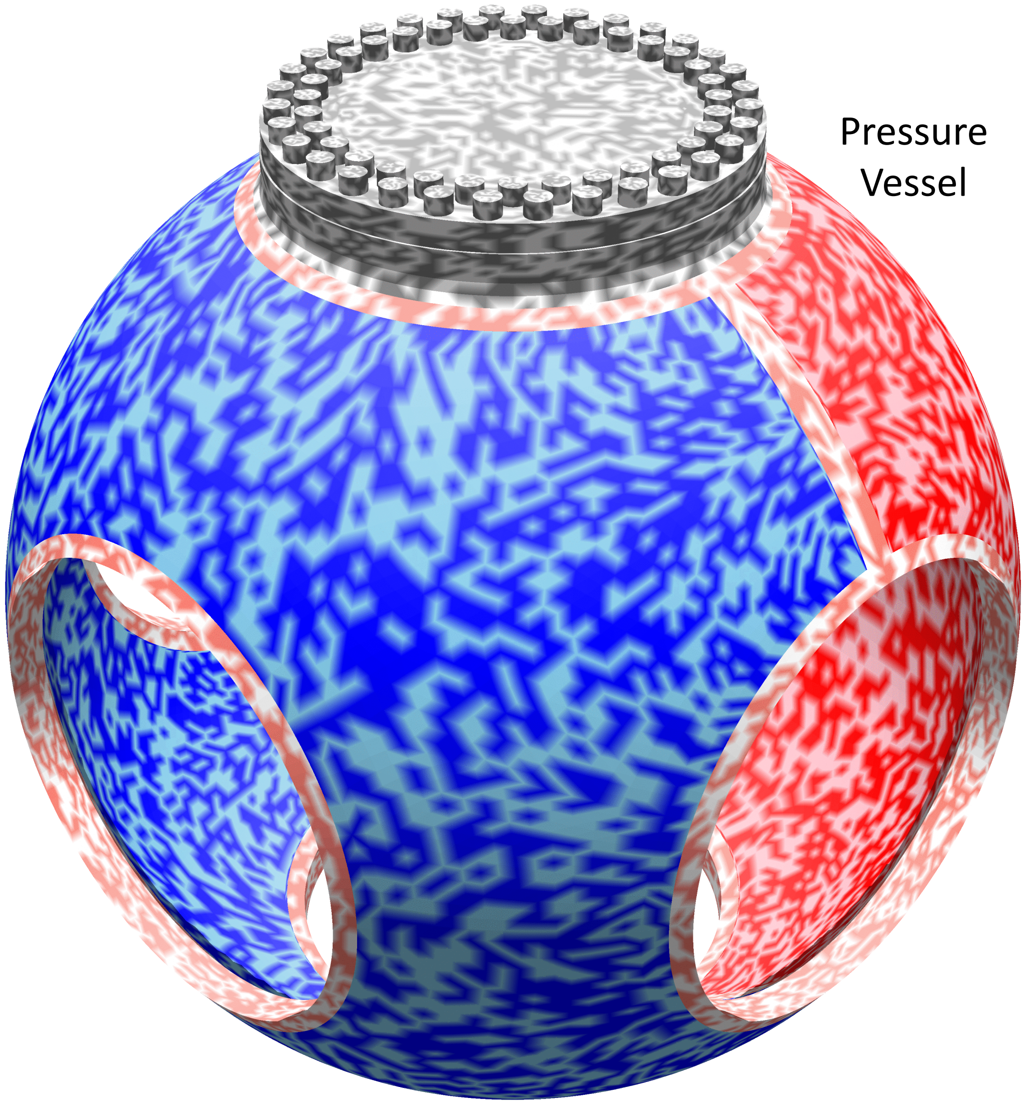
ABAQUS Meshes

DRACO can also parse input decks describing meshed objects for the ABAQUS finite element code.

This capability was desirable to allow the sharing of common meshes between groups working on different aspects of a problem.





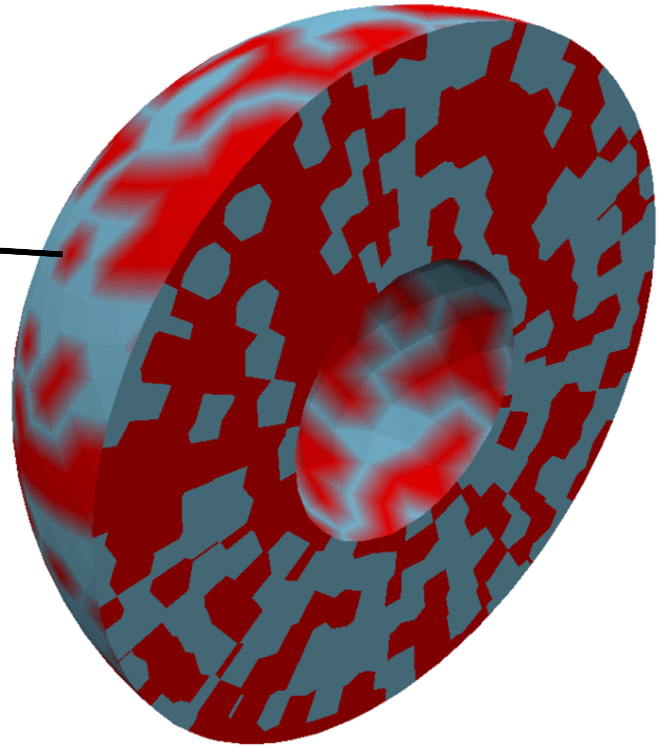


Pressure Vessel

Example objects rendered by DRACO from ABAQUS meshes

Surfaces

Surfaces of parts are defined and represented in DRACO via a triangular surface mesh.



Each triangle's corners are surface elements of the part. In image rendering, coloration, corresponding to temperature, concentration, etc., is determined via linear interpolation using barycentric coordinates.

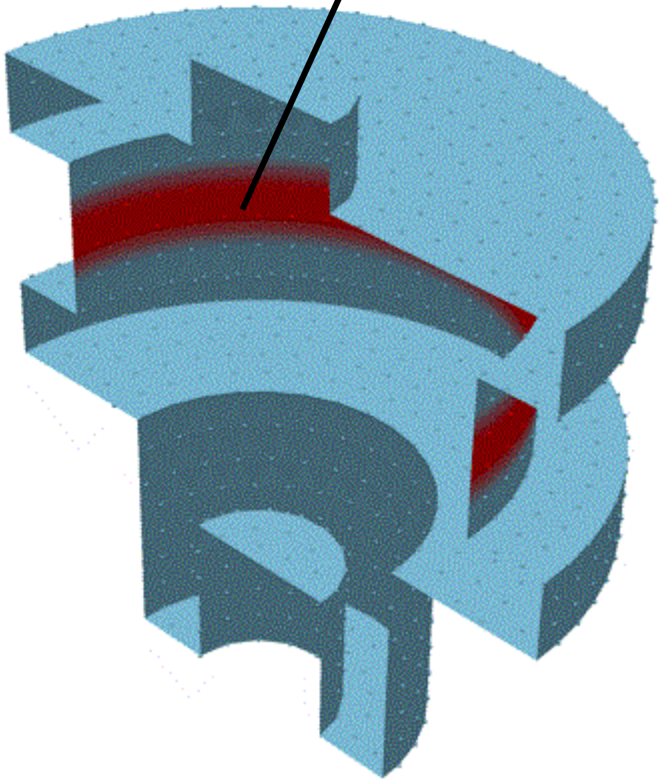
This surface representation also allows the construction of parts that consist **solely of surfaces**, i.e. they are 2D manifolds.

Such surfaces can be used to capture processes that occur only on material surfaces, or in gaps between objects.

Variable Surface Thickness

- Surface Diffusion with Variable “Skin Thickness”

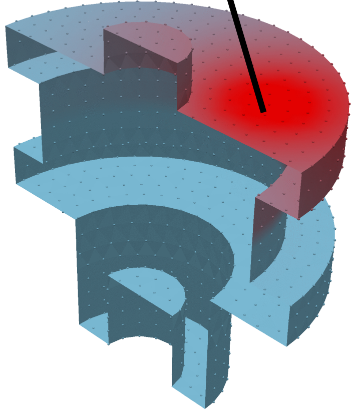
Shown here: A radially-symmetric test surface object. The skin thickness, where diffusion occurs, is **constricted** in the red-shaded region to 1/50th of its value elsewhere.



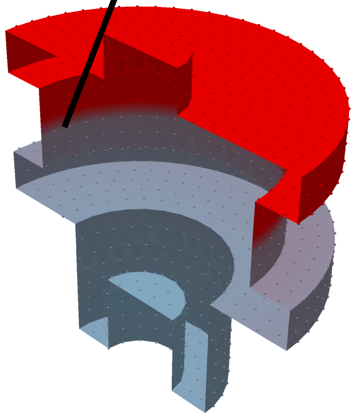
(Cutaway view showing the interior)

A source spot in the form of a circular patch is here

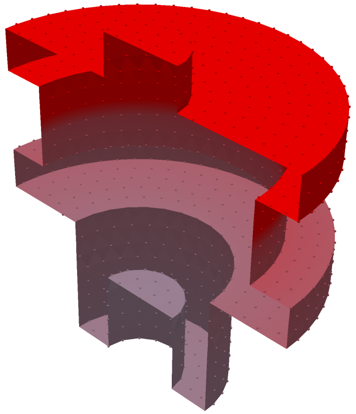
The diffusion wave of higher concentration takes much longer to pass by the constriction



1,000 time steps



6,000 time steps



10,000 time steps

15,000 time steps

Input Decks

DRACO reads in input decks provided by the user to specify all the parameters necessary to perform a simulation. This includes everything about:

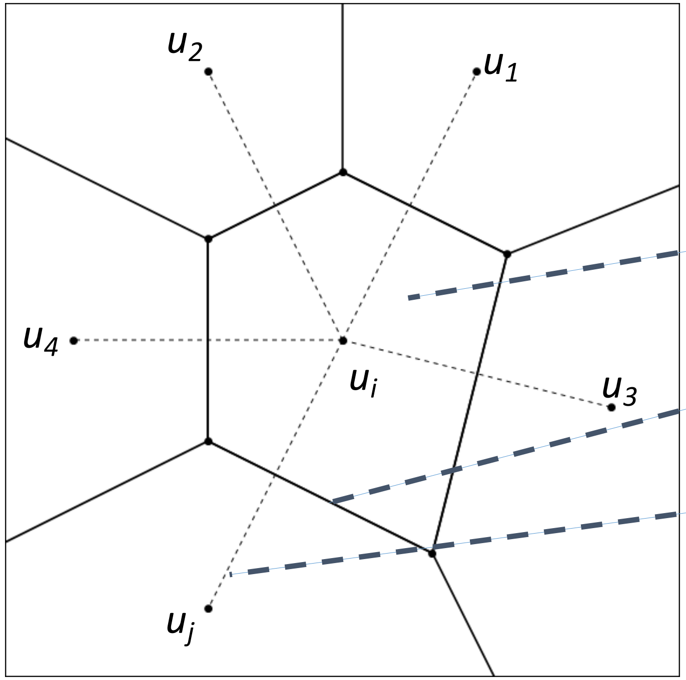
- The number of parts, part geometry, and mesh resolution.
- The nature and number of the diffusing/reacting species.
- Everything about image rendering and image/data output.
- Specification of arbitrary initial condition, boundary condition, and source functions. These can depend on time, space, concentration, etc.
- Chemistry. Which species react with one another in what quantities, at what rates, and with what products.

```

# Overall System parameters ##############################################################

IntroString This is a test DRACO simulation # Introductory statement to be printed near be

Verbose 0                        # Whether to report information about System and Parts "ve

NumParts 1                       # How many parts?
NumSpecies 0                     # How many diffusing/reacting species? (Must be >= 0. 0 im

#tmax 2.0                            # Maximum time to simulate to
tmax 0.0                             # Maximum time to simulate to
dt 0.001                             # Approximate timestep

ImageOnlyModeFlag 1              # If true (1), only an image of the initial condition will

SpeciesDrawList 0

ImageName ./Test/HorseshoeSurface            # Name of the image files to be output by the
ImageInterval 1                  # How many time steps between successive system images (<=
ImageFormat png
ImageWidth 2000                  # Width (in pixels) of the system images.
BackgroundColor White            # Background color of all system images

#ViewPoint 1.5 1.5 1.5                   # View Point for rendered system images in 3D
ViewPoint 0.0 3.0 2.0                    # View Point for rendered system images in 3D
#ViewPoint 2.5 0 0                   # View Point for rendered system images in 3D
#ViewPoint 0 3.0 0                   # View Point for rendered system images in 3D
LookPoint 0 0 0
#LookPoint 0 0 -1.0
LightDirection 0 0 -1
Up 0 0 1                         # Which direction shall be the "up" orientation for 3D ima
Angles 0.7 0.6                   # Vertical and horizontal angles (in radians) of the "came

```

Example Input Deck Snippet

Under the Hood

DRACO operates by dividing up the volume of each part into a **Voronoi tessellation** around the set of grid points:



A Voronoi tessellation divides a volume into "cells" around each point, where each point's cell is the region of closer to that point than any other.

This is advantageous because the **diffusion operator** can be represented in terms of the tessellation geometry:

$$\vec{\nabla} \cdot \left(D \vec{\nabla} u \right)_i \approx \frac{1}{V_i} \sum_j A_{ij}\, D_{ij}\, \frac{u_j - u_i}{d_{ij}}$$

Sum runs over all neighbors *j* of cell *i*

V_i is the <u>volume</u> of cell *i*

A_{ij} is the <u>area</u> of the facet between neighboring cells *i* and *j*

d_{ij} is the <u>distance</u> between cells *i* and *j*

$D_{ij} = \dfrac{2}{1/D_i + 1/D_j}$ is the <u>harmonic mean</u>* of D_i and D_j

$u_i - u_j$ is the concentration (or temperature, pressure,…) difference between cells *i* and *j*

* "An Integrated Finite Difference Method for Analyzing Fluid Flow in Porous Media," T. N. Narasimhan and P. A. Witherspoon, *Water Resources Research*, **12**(1) 57-64 (1976).

Stability and Validation

Diffusion in the DRACO integration scheme is **stable** or **unstable** depending on the value of the **CFL number**:

$$CFL = \max_{ij} \frac{\Delta t \, A_{ij} \, D_{ij}}{2 \, d_{ij}} \left(\frac{1}{V_i} + \frac{1}{V_j} \right) \propto \frac{D \Delta t}{\Delta x^2}$$

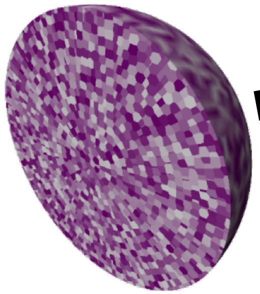
Timestep

Resolution

The CFL number must be below a certain threshold for stability. The exact threshold depends on the shape of the part and the nature of the mesh. For instance, for diffusion of random data in a sphere, by performing calculations at many values of the time step *Δt* and resolution *Δx*, we find that the CFL number and stability condition for a sphere are:

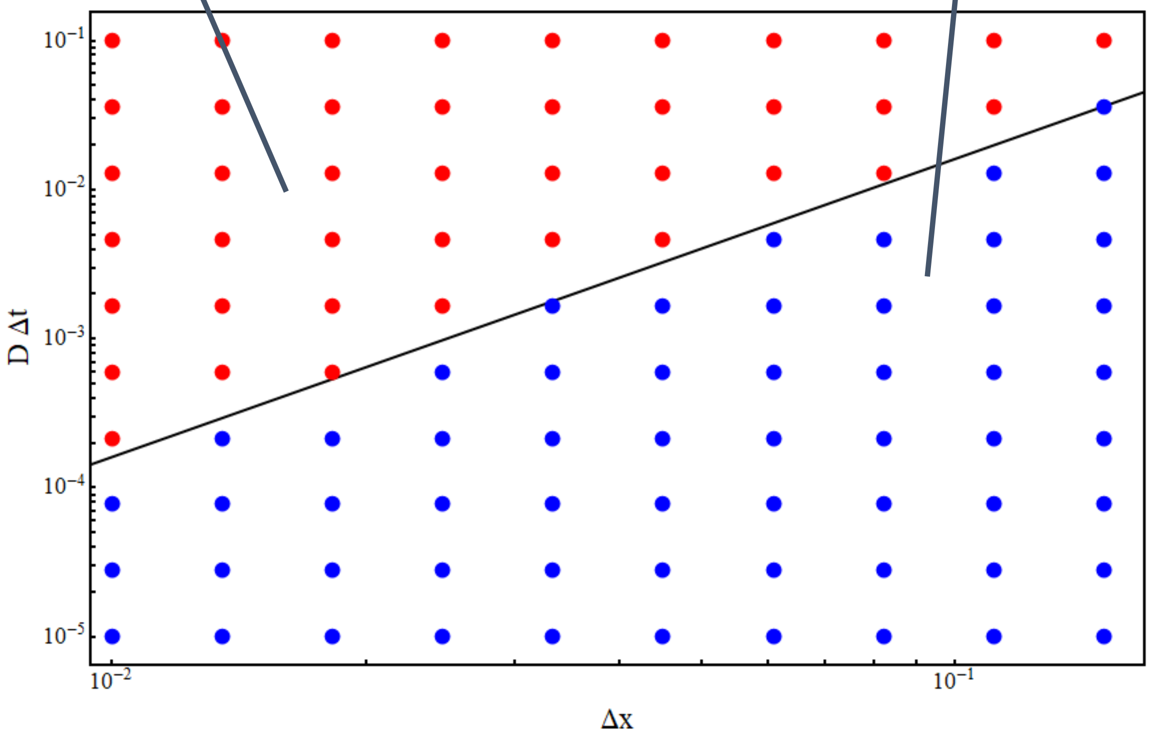
$$CFL \approx 1.5 \frac{D \, \Delta t}{\Delta x^2} < 0.31$$

Random Data in a Sphere



Red Points: Unstable Simulations

Blue Points: Stable Simulations



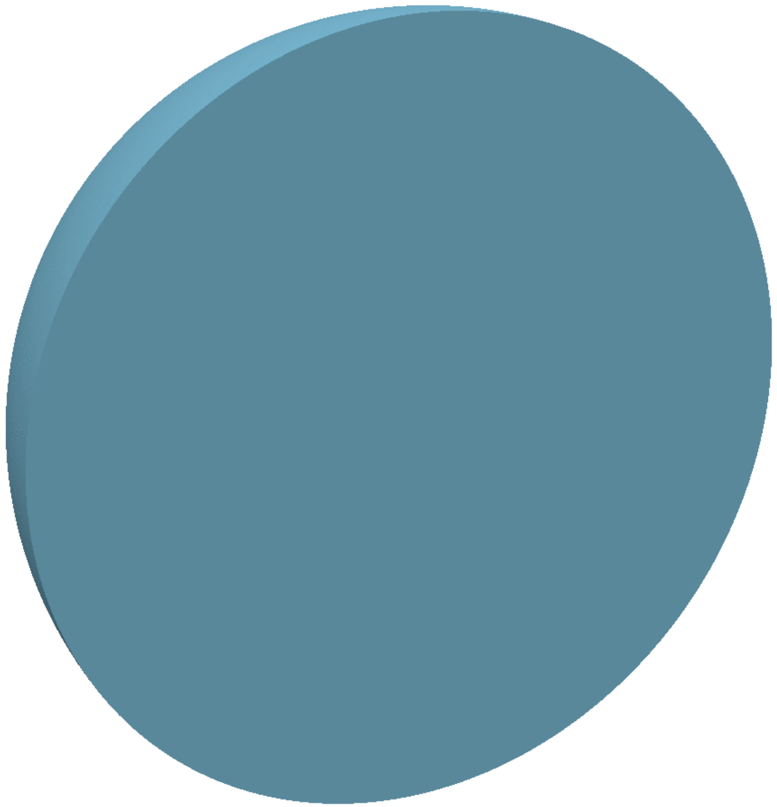
One way DRACO has been validated is by comparison in problems which have an **exact analytic solution**:

Example: Diffusion in a sphere of radius *R*:

$$\begin{cases} \dfrac{\partial}{\partial t} u(\vec{r},t) = D\,\nabla^2 u(\vec{r},t) & \\ u(\vec{r},0) = 0 & \text{Initial Condition} \\ u(\vec{r},t) = \underbrace{C}_{\text{Constant}}\,\underbrace{Y_\ell^m(\theta,\phi)}_{\text{Spherical Harmonic}} & \text{On Boundary} \end{cases}$$

$$u(\vec{r},t) = C\,Y_\ell^m(\theta,\phi)\left\{\left(\frac{r}{R}\right)^\ell - 2\left(\frac{R}{r}\right)^{1/2}\sum_{n=1}^{\infty}\frac{\exp\left[-D(x_{\ell n}/R)^2 t\right]}{x_{\ell n}\,J_{\ell+3/2}(x_{\ell n})}J_{\ell+1/2}\left(x_{\ell n}\frac{r}{R}\right)\right\}$$

DRACO simulation of this problem:



Cutaway View

$R = 0.5$

$\Delta t = 10^{-5}$

$\Delta x = 10^{-2}$

$D = 1$

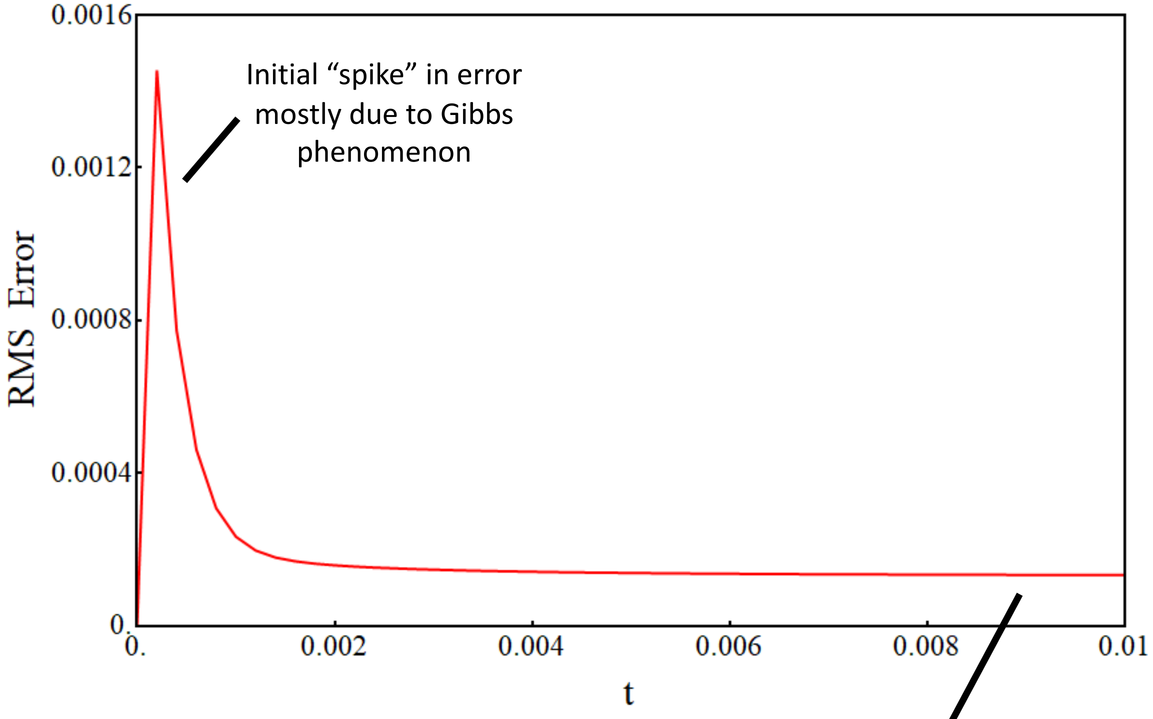
$CFL \approx 0.13$

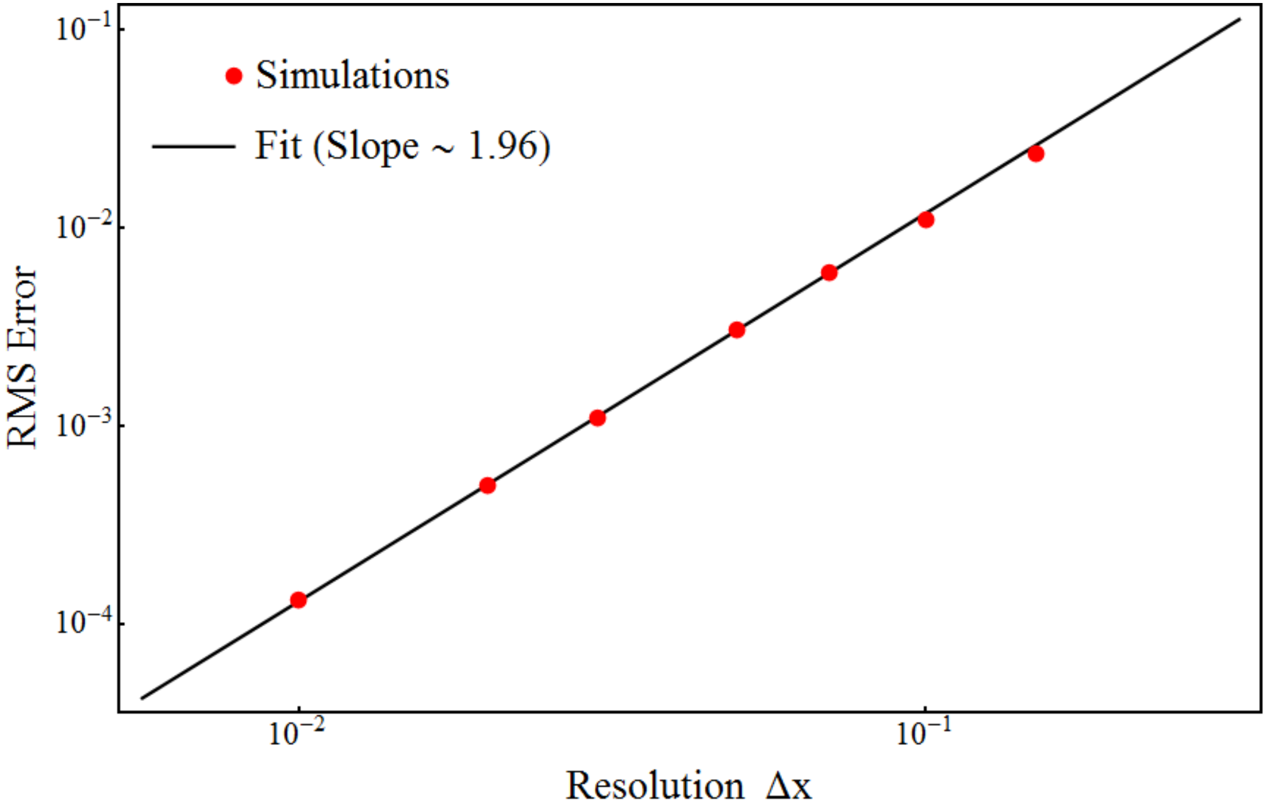
$\ell = 5$

$m = 3$

539,383 Elements

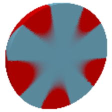
RMS error vs. time for an example simulation:





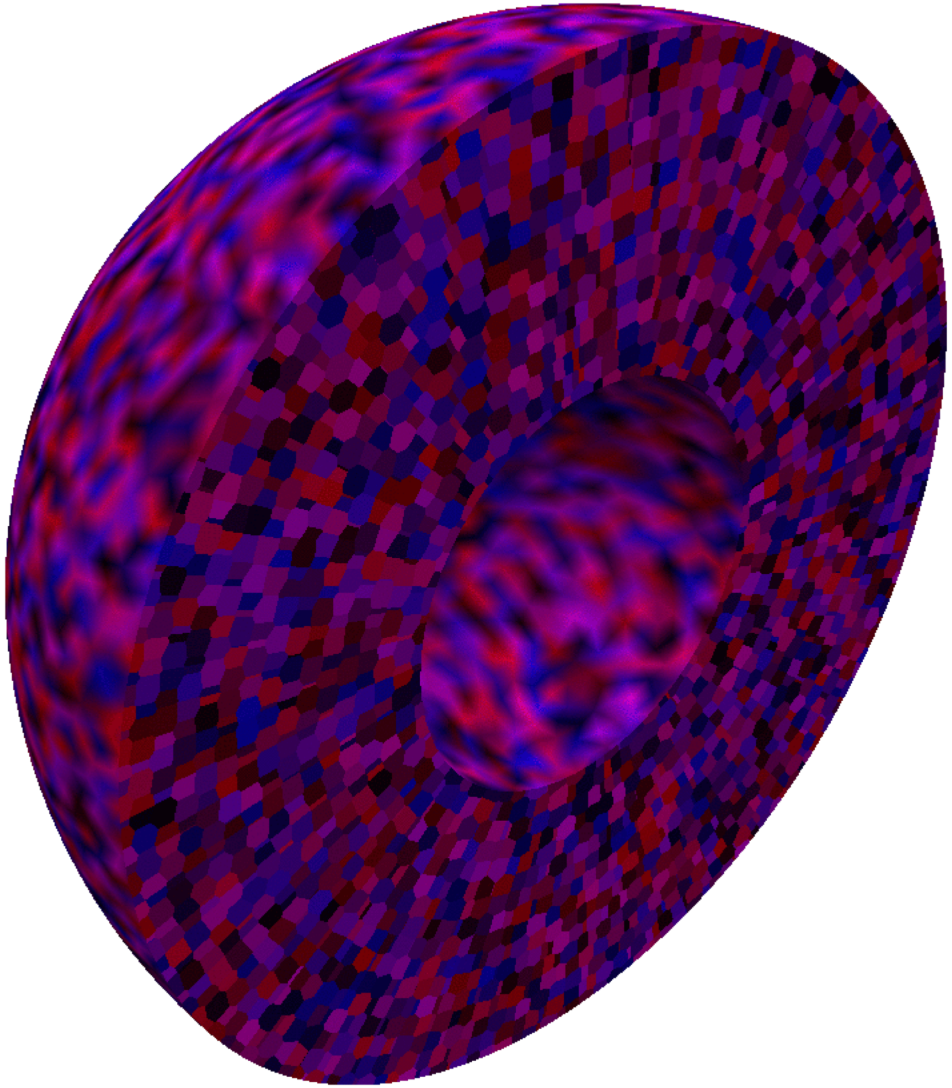
As t → ∞, we recover the RMS error with respect to the equilibrium solution.

$$u_\infty(\vec{r}) = C\, Y_\ell^m(\theta,\phi)\left(\frac{r}{R}\right)^\ell$$



Simulations at various resolutions *Δx* can be used to check the **scaling** of the error with *Δx*. As expected, DRACO is 2nd order in *Δx*.

Chemistry Model



Red: X concentration
Blue: Y concentration

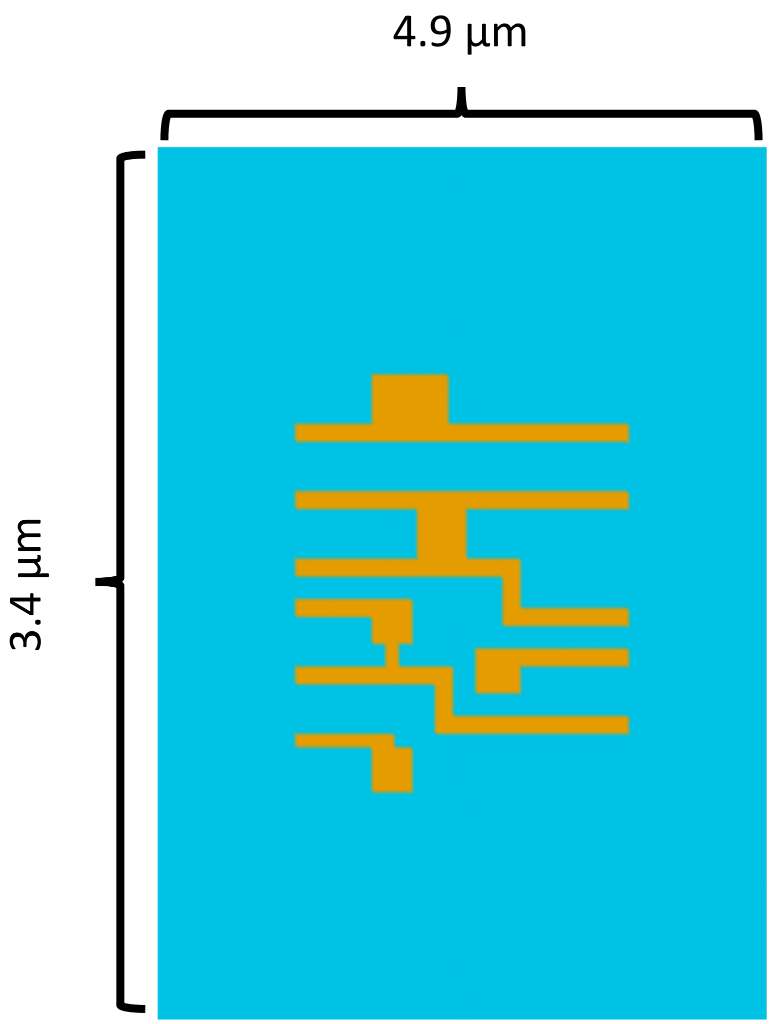
The “Brusselator” model of the **Belousov-Zhabotinsky oscillating chemical reaction**, with random initial conditions and boundary conditions, and simulated with in a spherical shell.

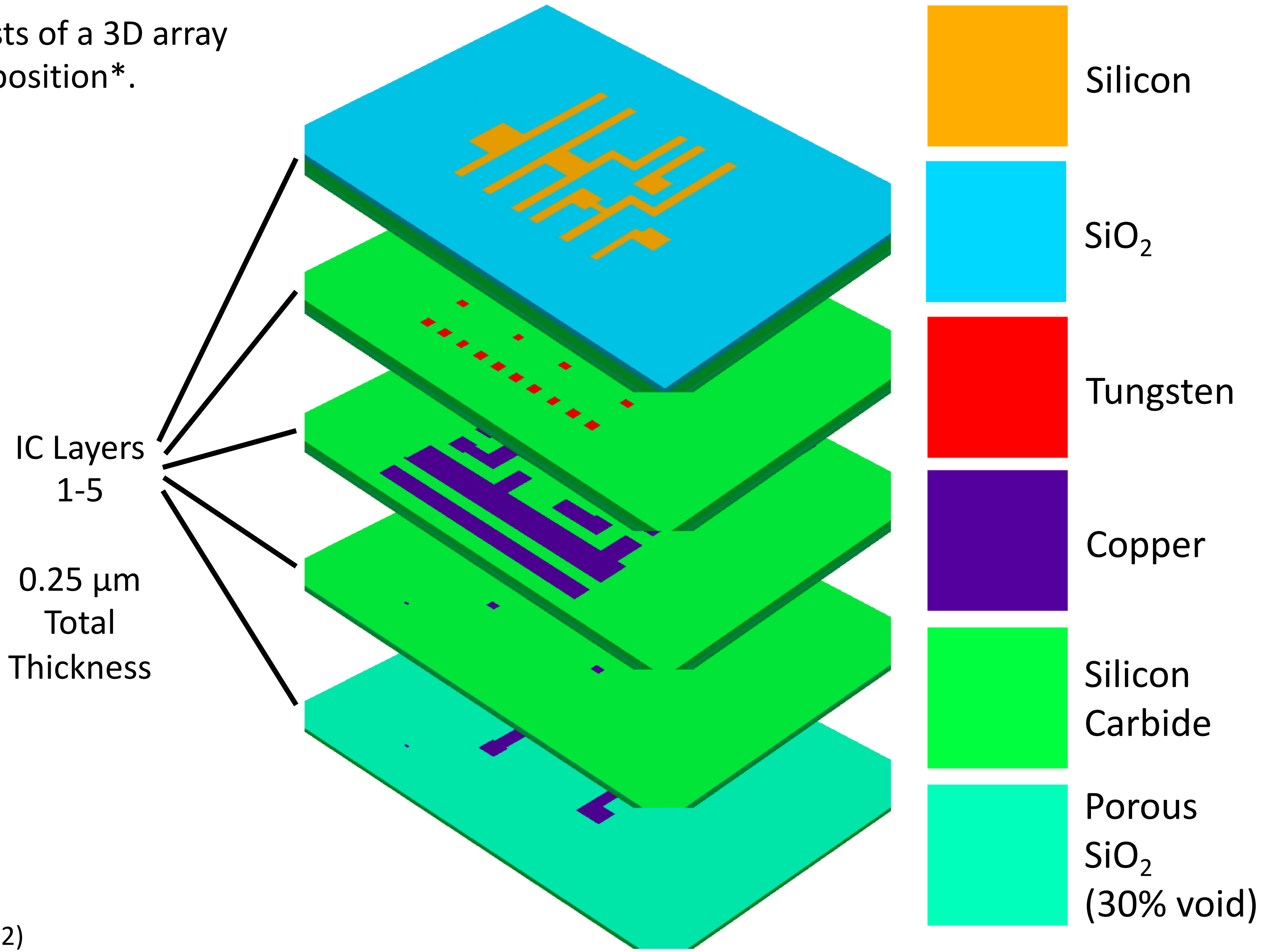
$$A \leftrightarrow X$$
$$2X + Y \leftrightarrow 3X$$
$$B + X \leftrightarrow Y + D$$
$$X \leftrightarrow E$$

Here the random boundary conditions are held fixed to provide seed points for **symmetry breaking**, leading to nonlinear diffusion waves that propagate through the system.

Dissipation of Laser-Induced Heating in Integrated Circuits

Model of an integrated circuit (IC) consists of a 3D array of voxels specifying the species at each position*.





*IC model data provided by Nina Weisse-Bernstein (ISR-2)

The integrated circuit is illuminated by a 20-100 femtosecond monochromatic laser pulse, which is modeled as having a uniform “top hat” profile. The **excess temperature** *ΔT(z)* deposited by the pulse at a depth *z* is then:

Number of Photons in pulse (4×10^6)

Photon Energy (10 keV)

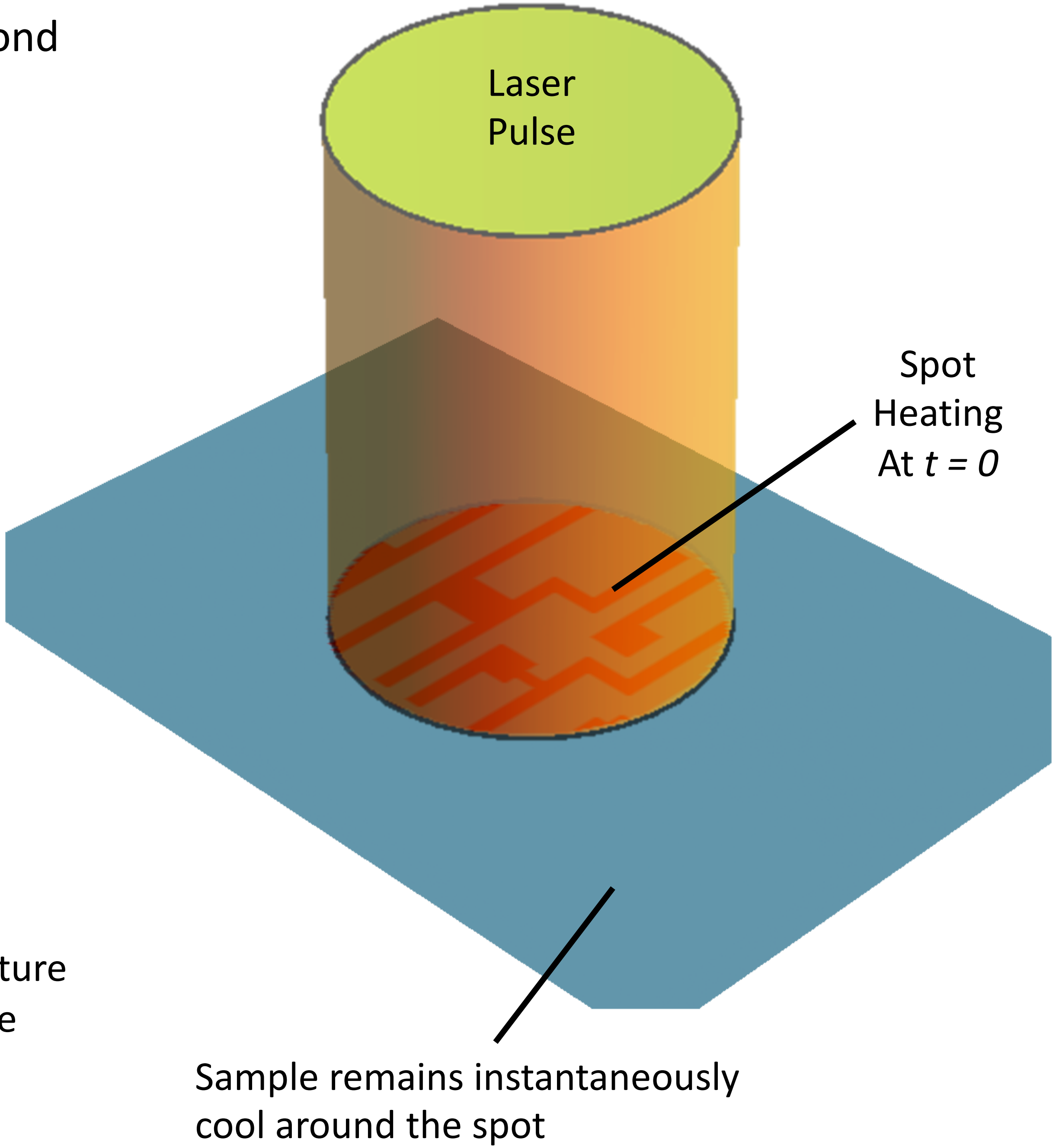
Total Absorption Coefficient

$$\Delta T(z) = \frac{4}{\pi}\,\frac{N_\gamma E_\gamma}{a^2}\,\frac{\mu(z)}{C_V(z)}\,\exp\left(\underbrace{-\int_0^z dz'\,\mu(z')}_{\text{Line Integral Of Absorption From Sample Surface To Depth } z}\right)$$

Spot Diameter (2 µm)

Volumetric Heat Capacity

The time scale (fs) of the pulse is **much shorter** than the time scale of heat dissipation by diffusion (ns), so this excess temperature can be treated as if deposited **instantaneously** at *t = 0*. The above profile is used as the initial condition for these simulations.



Simulation Details:

Resolution:

$\Delta x \approx 25$ nanometers

Time step:

$\Delta t \approx 0.5$ picoseconds

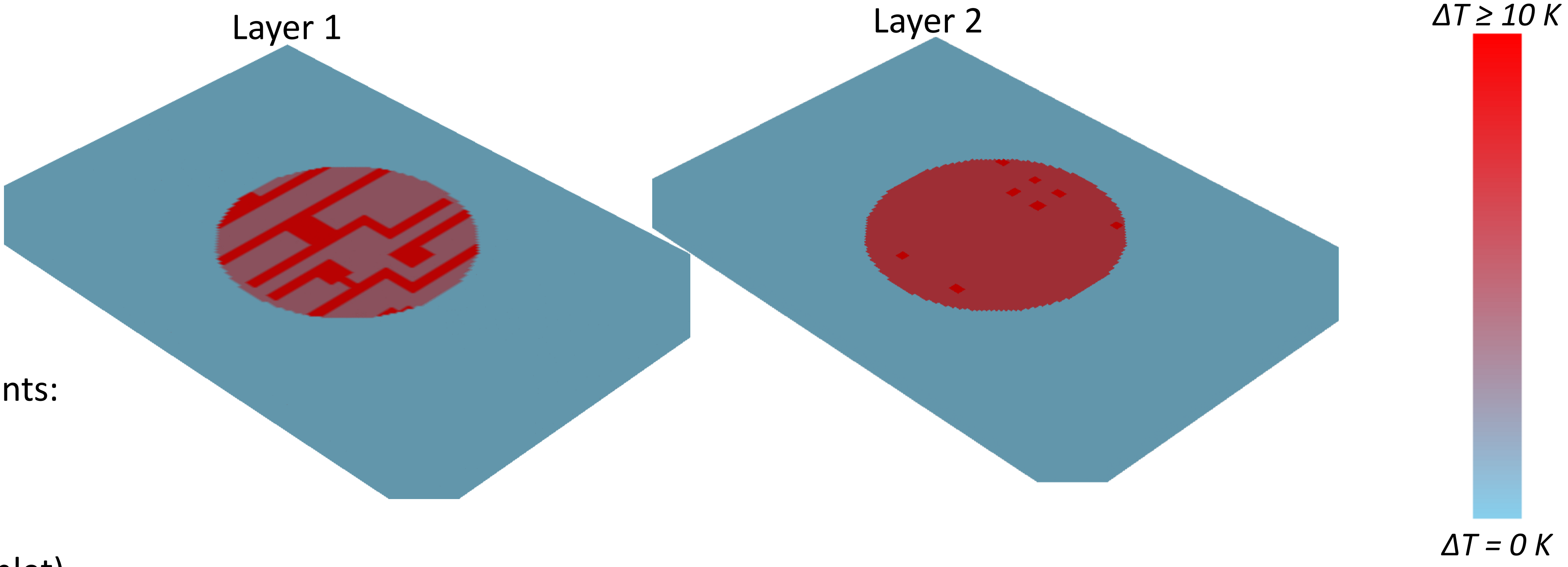
Number of DRACO Elements:

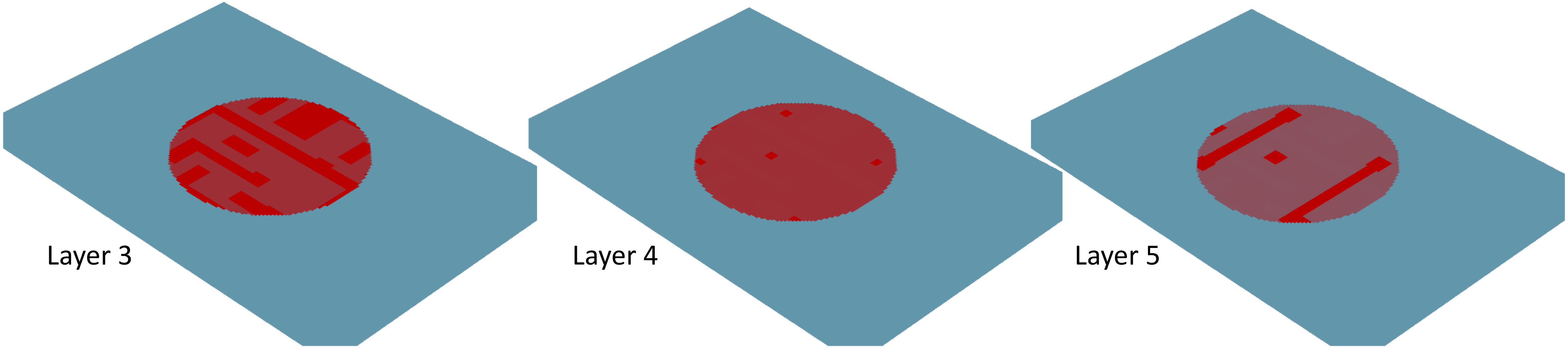
291,720

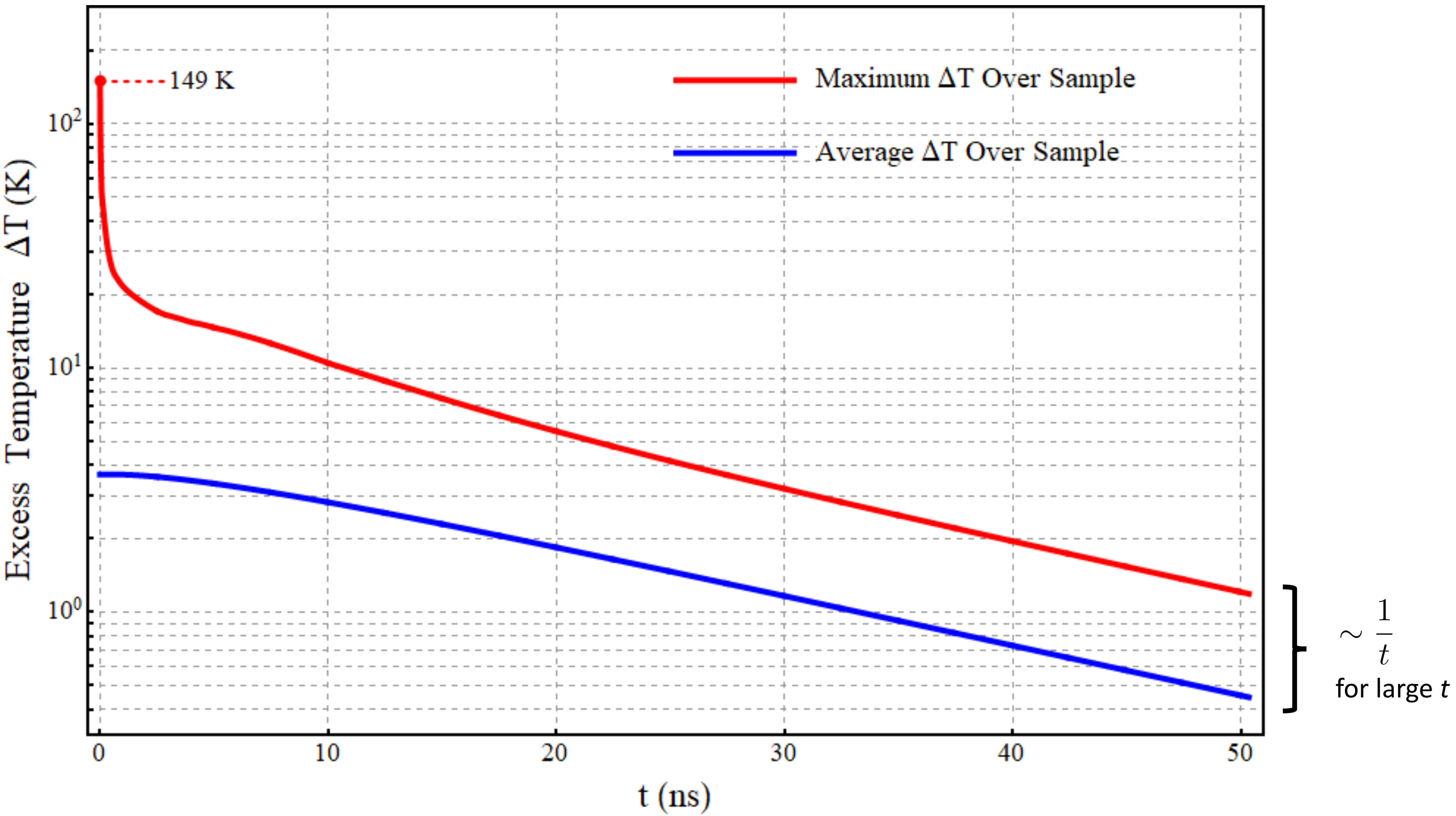
Boundary Conditions:

$\Delta T = 0$ On edges (Dirichlet)

Insulated on top/bottom (Neumann)






149 K
Maximum ΔT Over Sample
Average ΔT Over Sample
Excess Temperature ΔT (K)
10^2
10^1
10^0
0
10
20
30
40
50
t (ns)
$\sim \frac{1}{t}$
for large t